

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended) In a Java computing environment, a method of invoking a native method written in a native programming language without calling a Java Native Interface (JNI), said method comprising:

providing a set of macro instructions which are written in said native programming language with a direct reference to one or more Java parameters stored on a Java execution stack, wherein said one or more Java parameters are associated with said native method;

converting, by said set of macro instructions, said one or more Java parameters into at least one native parameter suitable for invocation of said native method, wherein said one or more Java parameters are converted by said converting without calling a Java Native Interface (JNI);

placing said at least one native parameter on a native execution stack associated with said native programming language after said converting operates to generate ing one or more said at least one native parameter[[s]] based on by accessing said direct reference; and

invoking said native method with said one or more at least one native parameter [[s]] after said converting of said at least one native parameter and said placing of said at least one native parameter on said execution stack, thereby allowing said native method to be invoked without calling said Java Native Interface (JNI).

2-3. (Cancelled)

4. (Currently Amended) A method as recited in claim [[3]] 1, wherein said set of macro instructions and said native method are written in C programming language.

5-8. (Cancelled)

9. (Original) A method as recited in claim 1, wherein said at least one Java parameter is a Java reference object or a Java primitive data type.

10-11. (Cancelled)

12. (Currently Amended) A method as recited in claim ~~[[11]]~~ 1,  
~~wherein said reference is provided directly to said set of macros; and~~  
wherein said set of macros operate to insulate the native method from the  
internals of a virtual machine that is invoking said native method.

13-30. (Cancelled)

31. (New) A computer readable medium including computer program code for  
invoking a native method written in a native programming language without calling a  
Java Native Interface (JNI), said compute readable medium comprising:

computer program code for providing a set of macro instructions which are  
written in said native programming language with a direct reference to one or more Java  
parameters stored on a Java execution stack, wherein said one or more Java  
parameters are associated with said native method;

computer program code for converting, by said set of macro instructions, said  
one or more Java parameters into at least one native parameter suitable for invocation  
of said native method, wherein said one or more Java parameters are converted by said  
converting without calling a Java Native Interface (JNI);

computer program code for placing said at least one native parameter on a  
native execution stack associated with said native programming language after said  
converting operates to generate said at least one native parameter by accessing said  
direct reference; and

computer program code for invoking said native method with said at least one  
native parameter after said converting of said at least one native parameter and said  
placing of said at least one native parameter on said execution stack, thereby allowing  
said native method to be invoked without calling said Java Native Interface (JNI).

32. (New) A computer readable medium as recited in claim 31, wherein said native programming language is the C or C++ programming language.

33. (New) A computer readable medium as recited in claim 31, wherein said at least one Java parameter is a Java reference object or a Java primitive data type.

34. (New) A computer readable medium as recited in claim 31, wherein said set of macros operate to insulate the native method from the internals of a virtual machine that is invoking said native method.

35. (New) A computer system, comprising:

- at least one processor which supports a virtual machine capable of invoking a native method written in a native programming language without calling a Java Native Interface (JNI), wherein said virtual machine is capable of:

- providing a set of macro instructions which are written in said native programming language with a direct reference to one or more Java parameters stored on a Java execution stack, wherein said one or more Java parameters are associated with said native method

- converting, by said set of macro instructions, said one or more Java parameters into at least one native parameter suitable for invocation of said native method, wherein said one or more Java parameters are converted by said converting without calling a Java Native Interface (JNI);

- placing said at least one native parameter on a native execution stack associated with said native programming language after said converting operates to generate said at least one native parameter by accessing said direct reference; and

- invoking said native method with said at least one native parameter after said converting of said at least one native parameter and said placing of said at least one native parameter on said execution stack, thereby allowing said native method to be invoked without calling said Java Native Interface (JNI).

36. (New) A computer system as recited in claim 35, wherein said native programming language is the C or C++ programming language.

37. (New) A computer system as recited in claim 35, wherein said at least one Java parameter is a Java reference object or a Java primitive data type.

38. (New) A computer system as recited in claim 35, wherein said set of macros operate to insulate the native method from the internals of said virtual machine.